Revisiting Resource Pooling: The Case for In-network Resource Sharing

Lorenzo Saino

Department of Electronic and Electrical Engineering
University College London

Joint work with Ioannis Psaras and George Pavlou

Outline

- Background
  - Resource pooling
  - Information Centric Networking
- In-network resource pooling
  - Main concepts
  - High level operation
- Early results
- Summary

Resource Pooling

Resource Pooling

- Well known principle in systems design supporting efficient utilization of resources under variable user demands.

Resource Pooling

- Well known principle in systems design supporting efficient utilization of resources under variable user demands.
- Resource pooling deeply routed in Internet architecture:
  - Packet switching enables pooling of link capacities and routers processing power
  - Buffers enable pooling of link capacity at adjacent time periods
  - MPLS traffic engineering and ECMP enable pooling of multiple paths

Resource Pooling

- Well known principle in systems design supporting efficient utilization of resources under variable user demands.
- Resource pooling deeply routed in Internet architecture:
  - Packet switching enables pooling of link capacities and routers processing power
  - Buffers enable pooling of link capacity at adjacent time periods
  - MPLS traffic engineering and ECMP enable pooling of multiple paths
- Pooled resources:
  - Router processing power
  - Links
  - Buffers
  - Paths

Efficiently Pooling End-to-end Paths

Efficiently Pooling End-to-end Paths

- Multipath TCP has been recently proposed to efficiently pool end-to-end paths

Efficiently Pooling End-to-end Paths

- Multipath TCP has been recently proposed to efficiently pool end-to-end paths
- Multiple simultaneous connections are opened between two communicating hosts over different paths

Efficiently Pooling End-to-end Paths

- Multipath TCP has been recently proposed to efficiently pool end-to-end paths
- Multiple simultaneous connections are opened between two communicating hosts over different paths
- Load is dynamically shifted among each path based on available bandwidth

Efficiently Pooling End-to-end Paths

- Multipath TCP has been recently proposed to efficiently pool end-to-end paths
- Multiple simultaneous connections are opened between two communicating hosts over different paths
- Load is dynamically shifted among each path based on available bandwidth
- Assumes that at least one host is multihomed

Efficiently Pooling End-to-end Paths

- Multipath TCP has been recently proposed to efficiently pool end-to-end paths
- Multiple simultaneous connections are opened between two communicating hosts over different paths
- Load is dynamically shifted among each path based on available bandwidth
- Assumes that at least one host is multihomed
- More reactive and fine-grained control than MPLS traffic engineering and ECMP

The long discussion on TCP

The long discussion on TCP

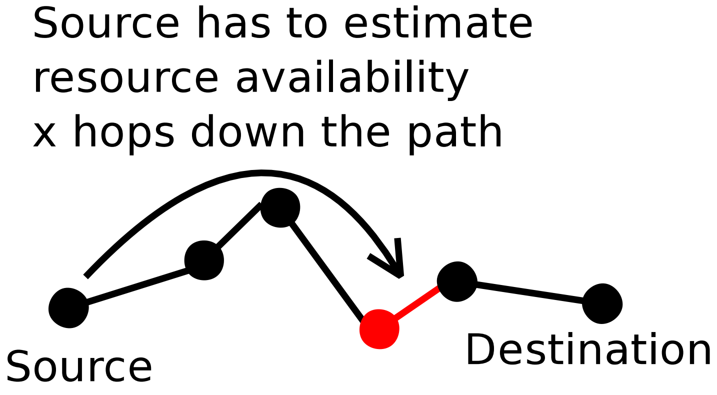
- TCP addresses uncertainty using the packet conservation principle and by (proactively) suppressing demand

The long discussion on TCP

- TCP addresses uncertainty using the packet conservation principle and by (proactively) suppressing demand
- TCP is moving traffic as fast as the path's slowest link

The long discussion on TCP

- TCP addresses uncertainty using the packet conservation principle and by (proactively) suppressing demand
- TCP is moving traffic as fast as the path's slowest link
- End-points have to speculate on the resources available along the end-to-end path



(i) e2e Resource Management

Information Centric Networking (ICN)

Information Centric Networking (ICN)

- ICN paradigm aims at rethinking Internet architecture having in mind content distribution as the main use case

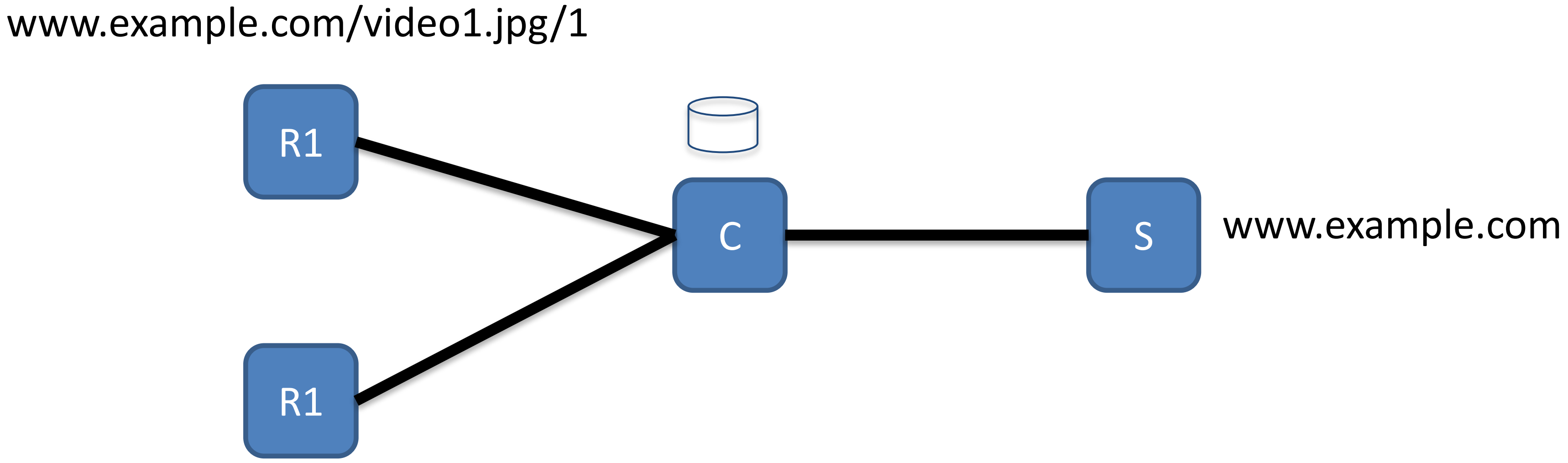
Information Centric Networking (ICN)

- ICN paradigm aims at rethinking Internet architecture having in mind content distribution as the main use case
- CCN/NDN [CoNEXT'09] is the most prominent architecture

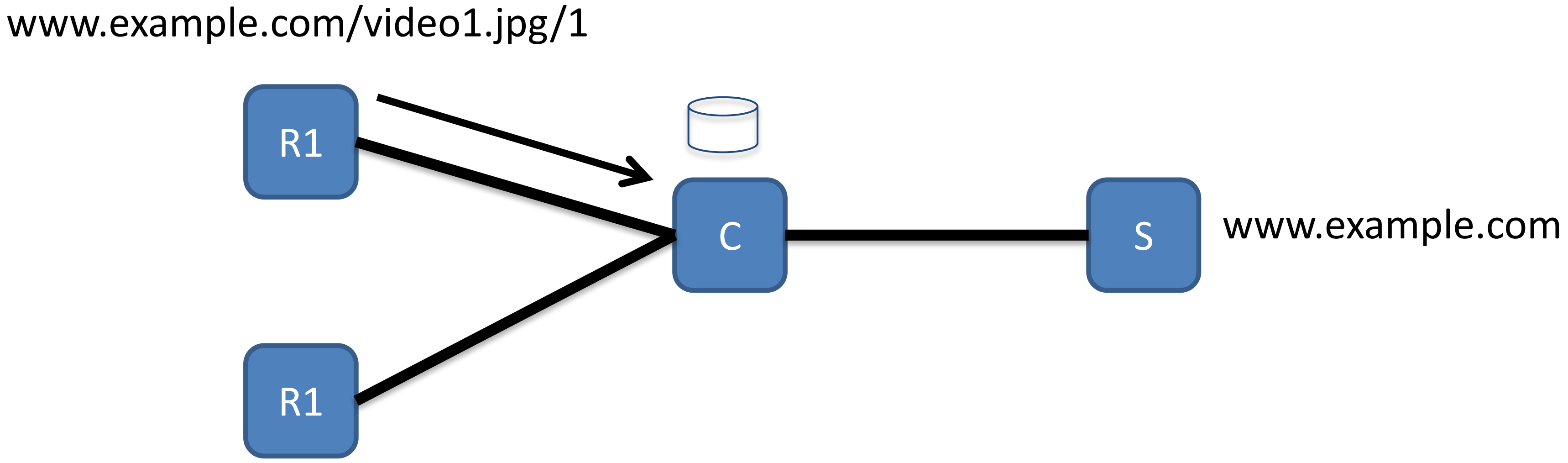
Information Centric Networking (ICN)

- ICN paradigm aims at rethinking Internet architecture having in mind content distribution as the main use case
- CCN/NDN [CoNEXT'09] is the most prominent architecture
- Main principles:
  - Naming contents instead of hosts
  - Receiver-driven request-response mode of operation
  - Securing content, not channel
  - Ubiquitous packet caches on routers

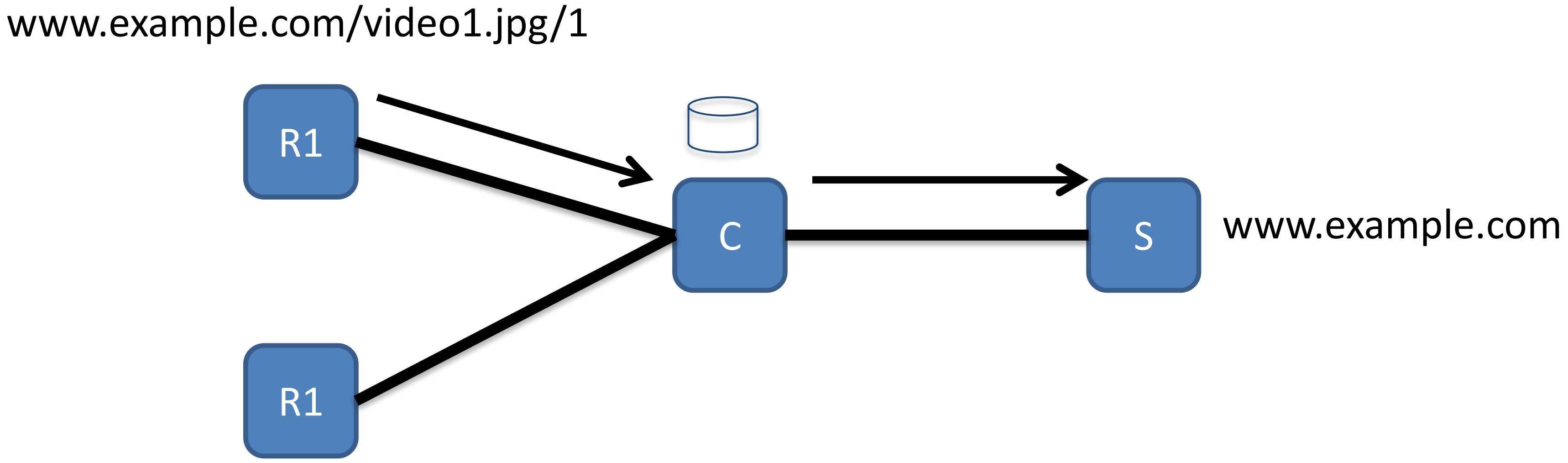
ICN operation



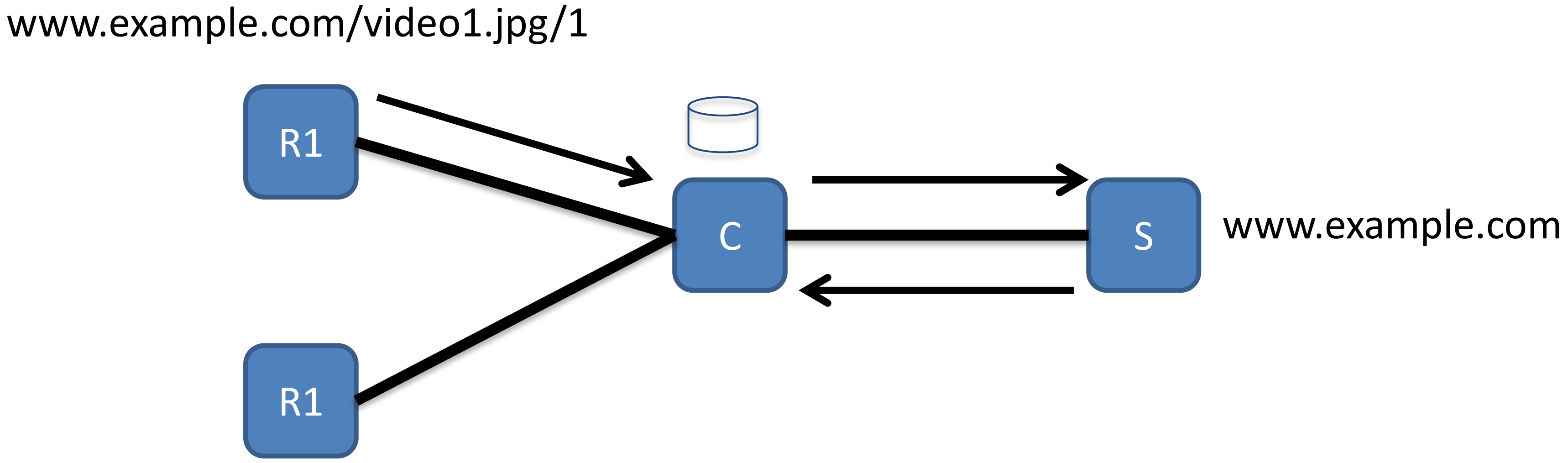
ICN operation



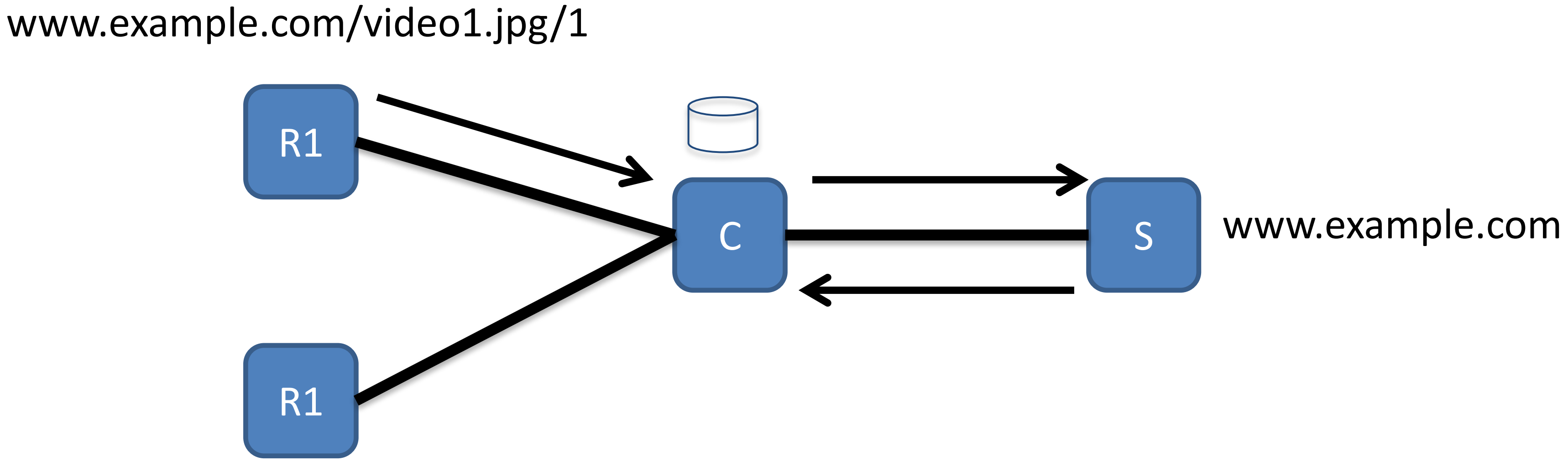
ICN operation



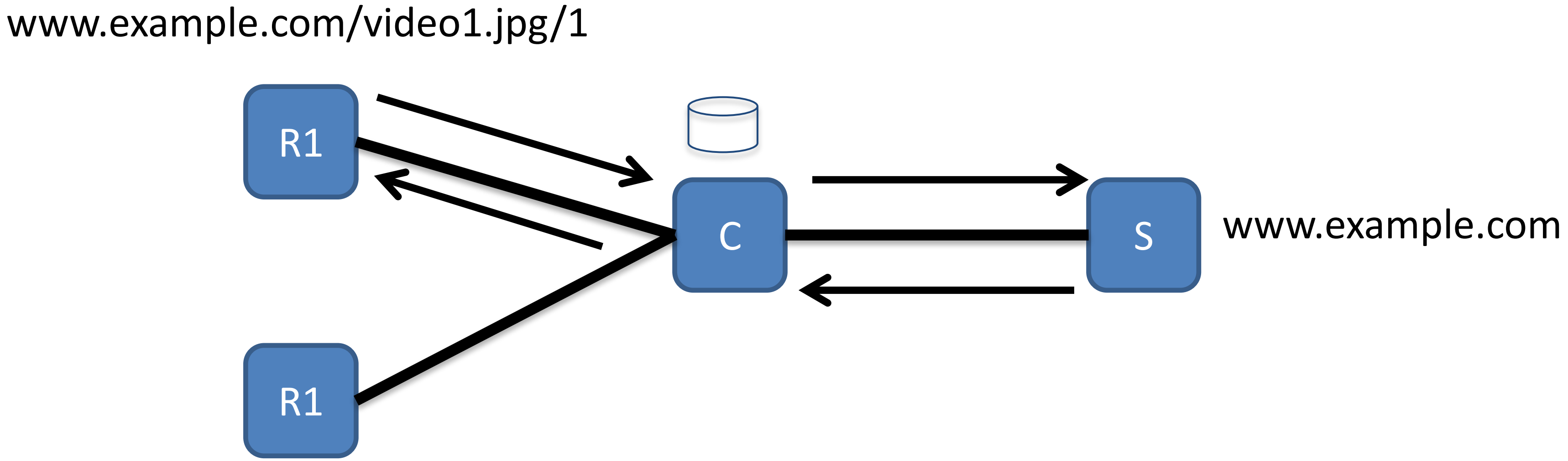
ICN operation



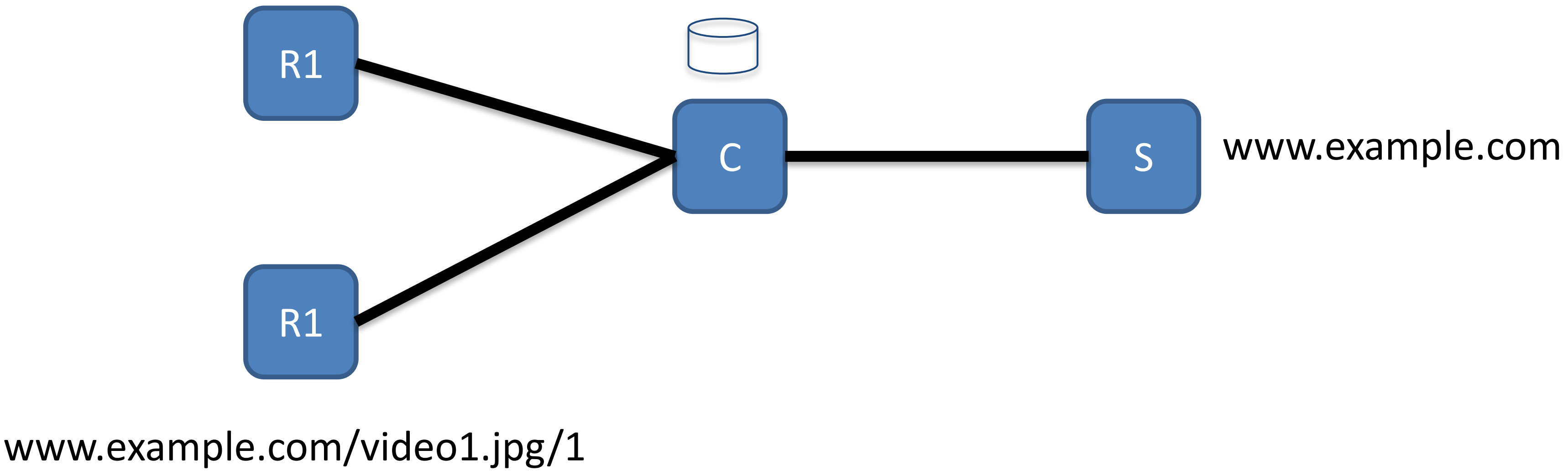
ICN operation



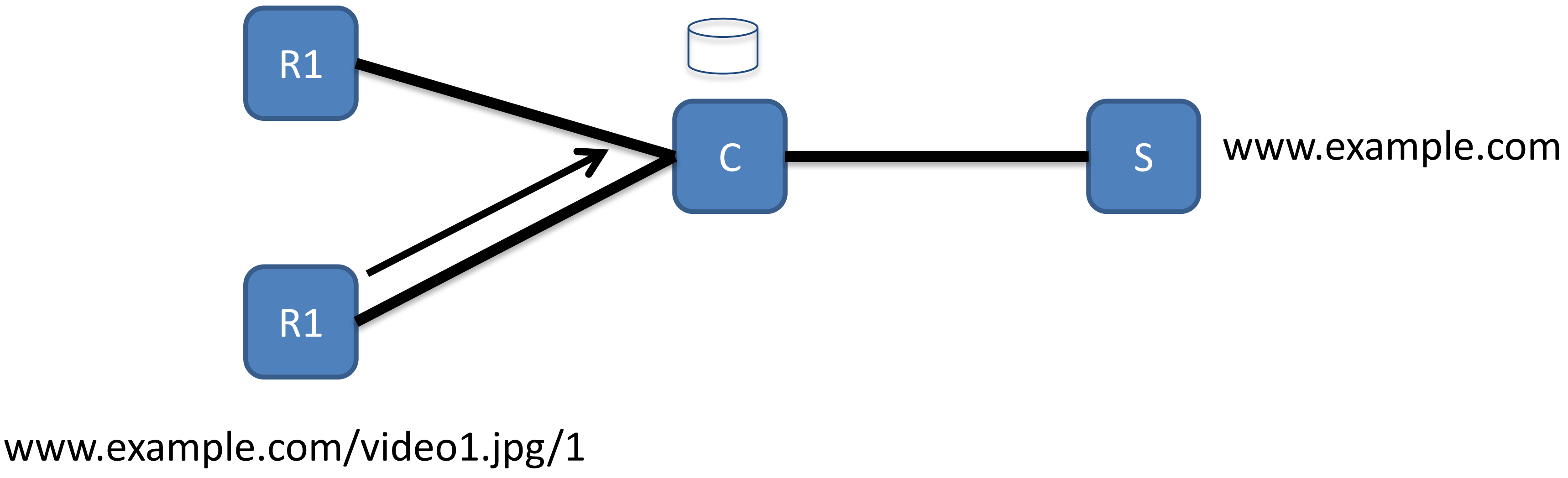
ICN operation



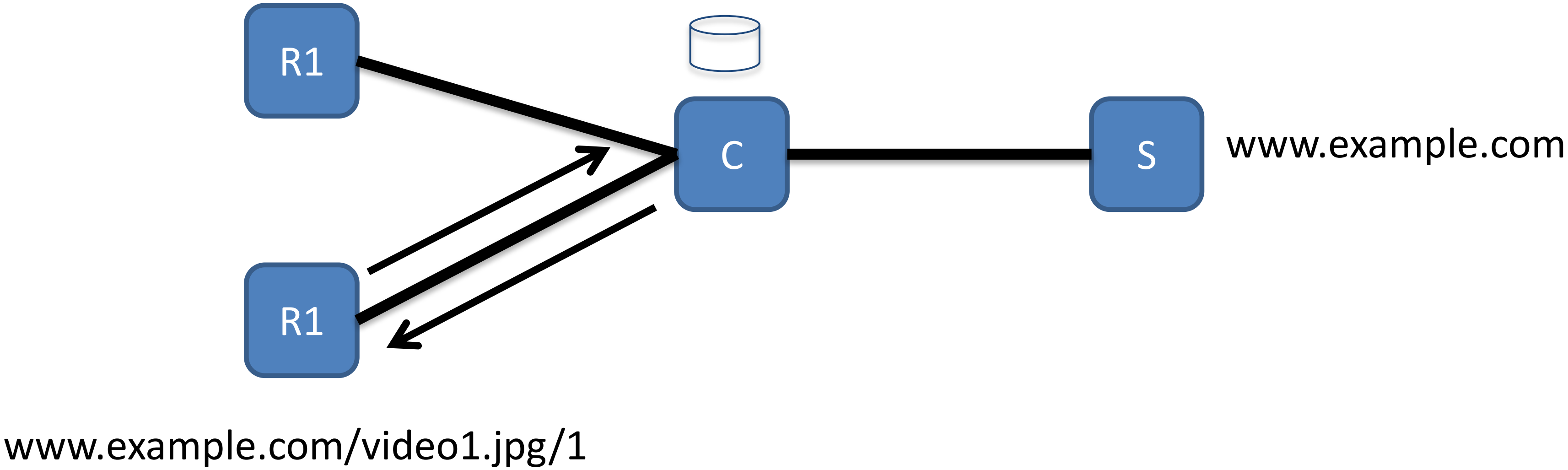
ICN operation



ICN operation



ICN operation



Transport layer implications of ICN

Transport layer implications of ICN

- Request and Data paths are symmetric
- Instead of the *"data-ACK"* model of TCP, in ICN we have a *"request-data"* model

Transport layer implications of ICN

- Request and Data paths are symmetric
- Instead of the *"data-ACK"* model of TCP, in ICN we have a *"request-data"* model

- Receivers (instead of senders) regulate the traffic that is pushed in the network
- Based on requests forwarded, each forwarding entity knows how much traffic to expect within one RTT.

In-network caches as resources

- Network caches have been used for *resource optimization* by storing popular contents, possibly for long time
  - Reduce latency, load on origin servers and bandwidth utilization
- Overlay caching:
  - Put caches in “strategic” places and redirect (HTTP) requests to those caches

In-network caches as resources

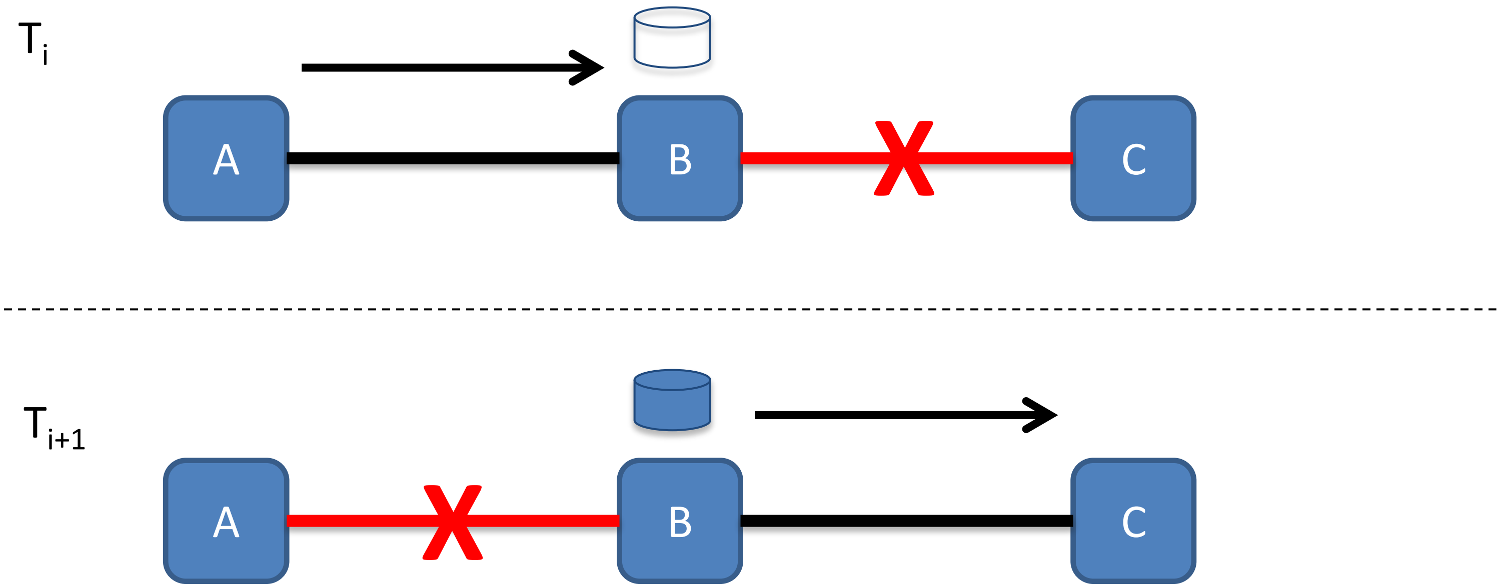
- Network caches have been used for *resource optimization* by storing popular contents, possibly for long time
  - Reduce latency, load on origin servers and bandwidth utilization
- Overlay caching:
  - Put caches in “strategic” places and redirect (HTTP) requests to those caches
- Packet caching:
  - Individually named and self-identifiable packets/chunks allow for in-network storage!
  - Put caches in every router and serve network-layer requests for named chunks from caches on the path

In-network caches as resources

- Network caches have been used for *resource optimization* by storing popular contents, possibly for long time
  - Reduce latency, load on origin servers and bandwidth utilization
- Overlay caching:
  - Put caches in "strategic" places and redirect (HTTP) requests to those caches
- Packet caching:
  - Individually named and self-identifiable packets/chunks allow for in-network storage!
  - Put caches in every router and serve network-layer requests for named chunks from caches on the path
- We use in-network caching for *temporary storage*

Caches and resource pooling

- The presence of ubiquitous packet caches enables more efficient usage of resources by enabling pooling of sub-paths.
- More effective than buffers



Pooled resources

Pooled resources

Links

Switching devices

Buffers

Packet switching

Pooled resources

Links

Switching devices

Buffers

Packet switching

Paths

ECMP, MPLS TE, MPTCP

Pooled resources

Links

Switching devices

Buffers

Packet switching

Paths

ECMP, MPLS TE, MPTCP

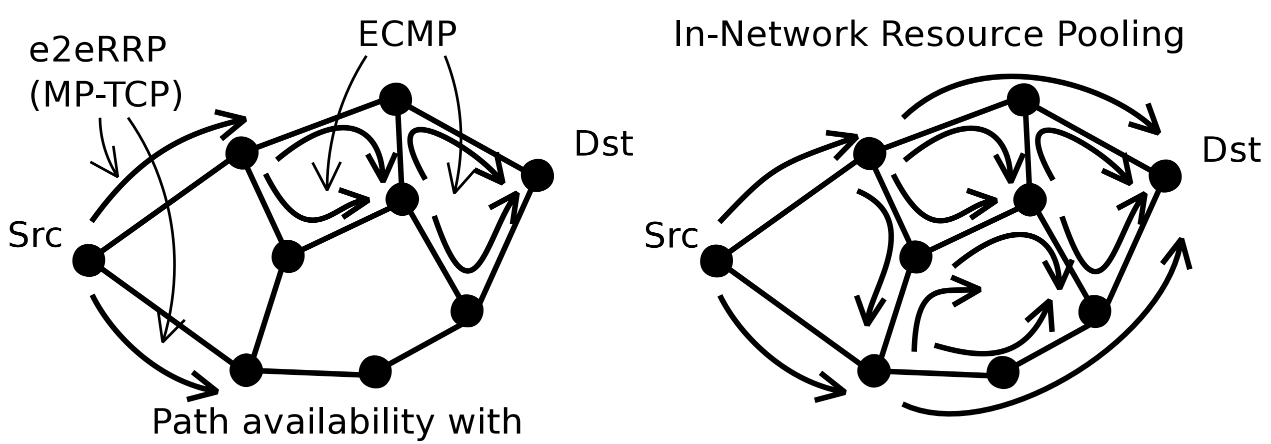
Sub-paths

Packet caches

Our proposal

Proposed solution

1. Push traffic *as far in the path and as fast* as possible
2. Once in front of the bottleneck, *store traffic temporarily* in custodian nodes/routers and deal with congestion locally
3. Exploit all available (sub-)paths making decisions on a *hop-by-hop manner*.



(ii) Path availability with e2e Resource Pooling and ECMP

(iii) Path Availability with INRP

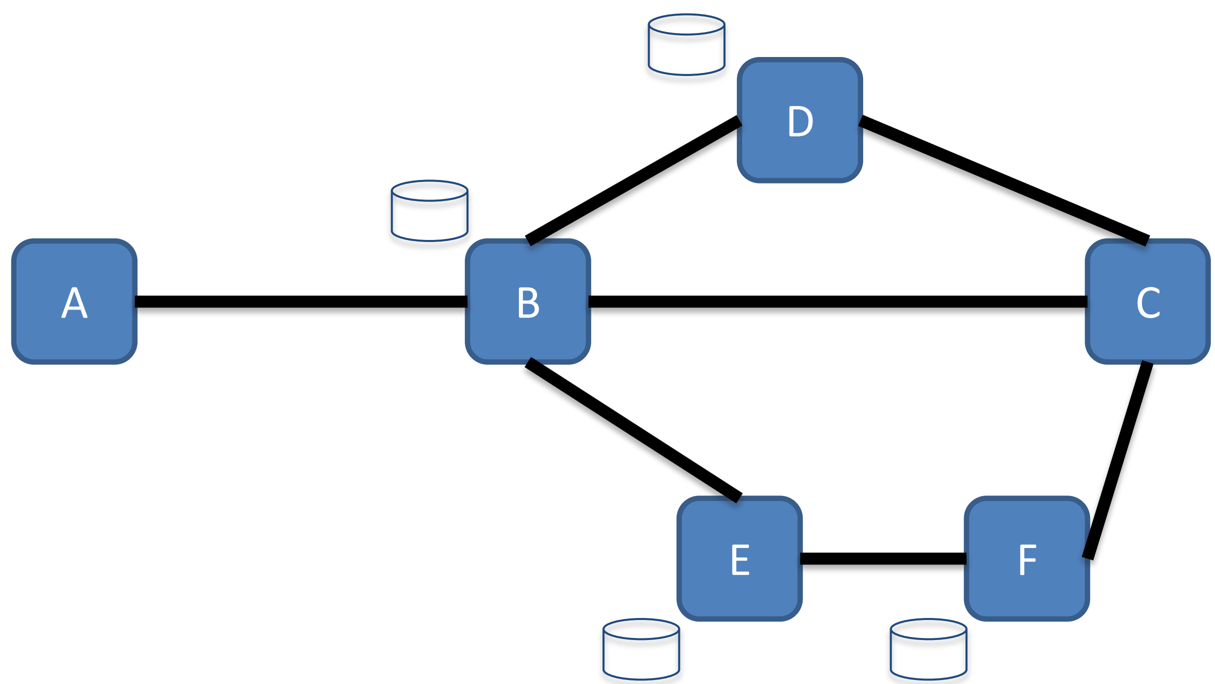
Three-phase operation

- **Push-data phase** – Open-loop system
  - Receivers request for as much data as supported by their access link
  - Senders push data as far and as quickly as possible
- **Cache & Detour phase**
  - Every router monitors rate of incoming *Requests*
  - When demand is expected to exceed supply, the local router tries to find alternative paths to detour
  - In the meantime traffic in excess (if any) is cached locally
- **Backpressure phase** – Closed-loop system
  - If alternative paths do not exist or are equally congested:
    - Pace requests
    - Send notification upstream to slow down and enter closed-loop transmission

Three-phase operation

Push-data phase – open-loop system

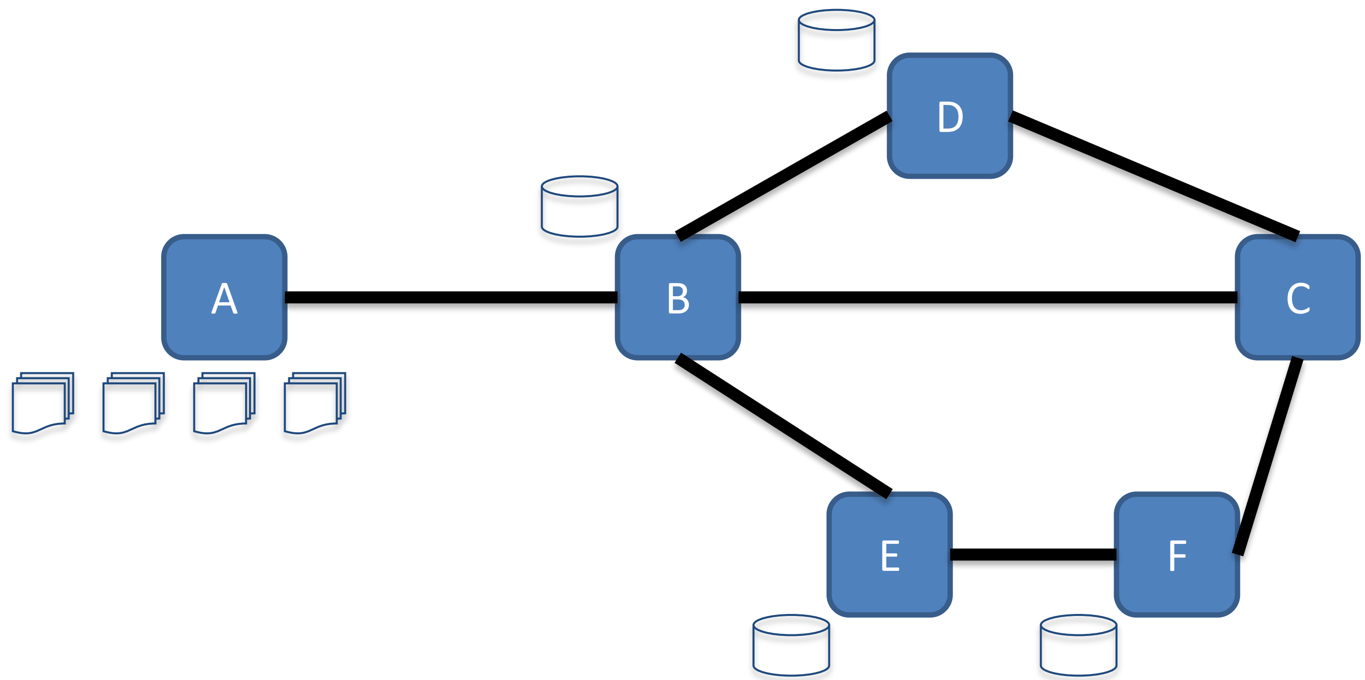
- Receivers request for as much data as supported by their access link
- Senders push data as far and as quickly as possible



Three-phase operation

Push-data phase – open-loop system

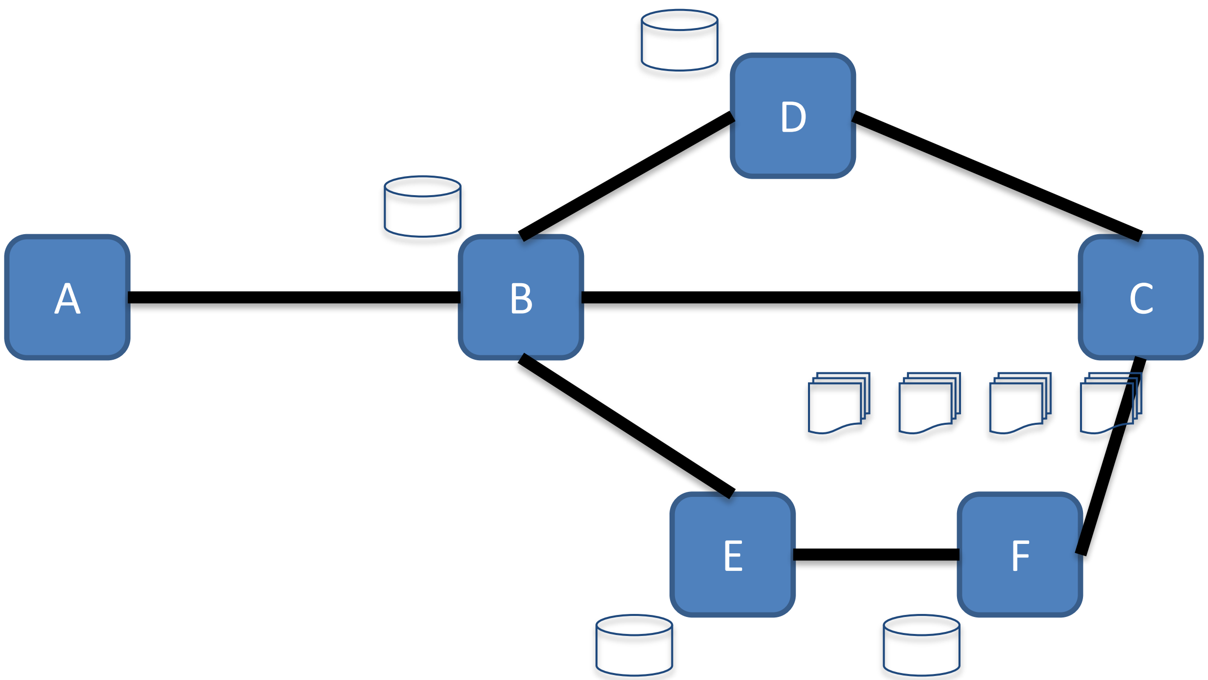
- Receivers request for as much data as supported by their access link
- Senders push data as far and as quickly as possible



Three-phase operation

Push-data phase – open-loop system

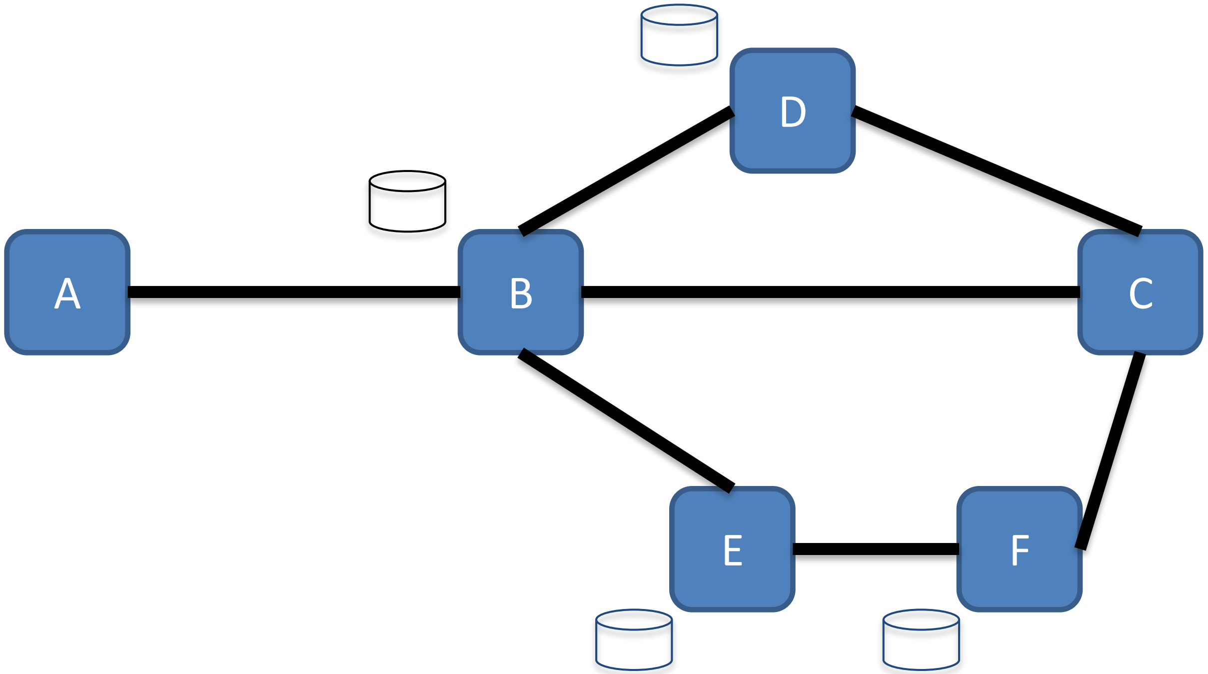
- Receivers request for as much data as supported by their access link
- Senders push data as far and as quickly as possible



Three-phase operation

Cache & Detour phase

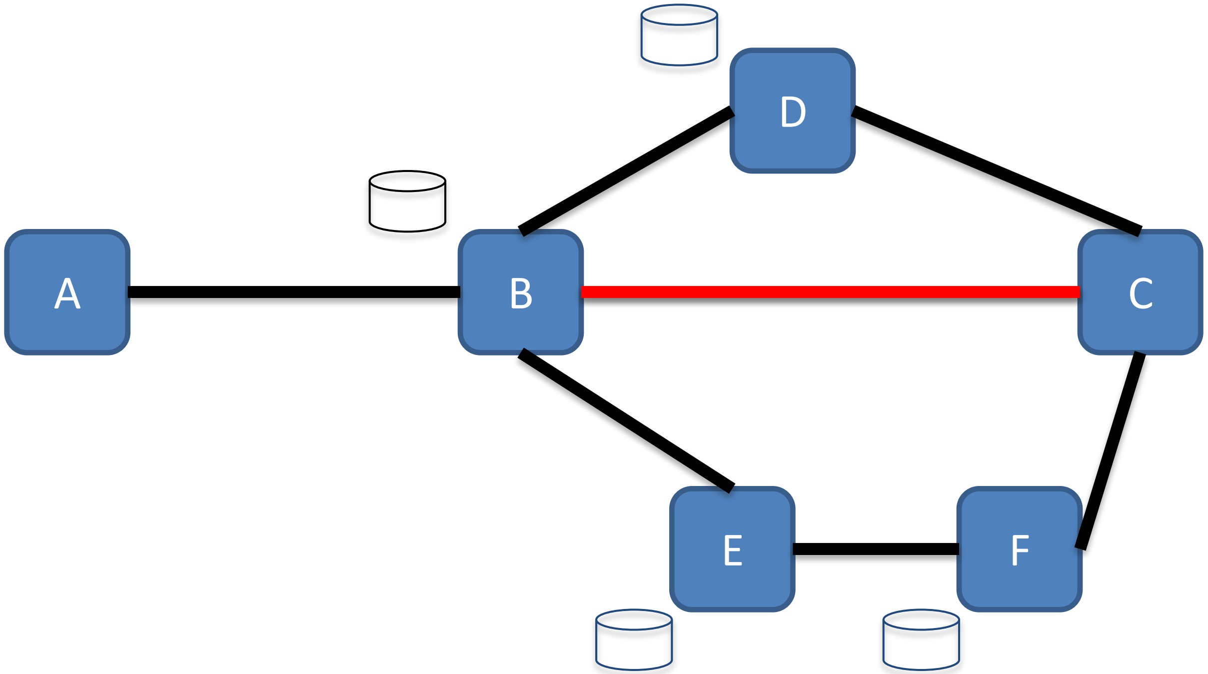
- Every router monitors rate of incoming *Requests*
- When demand is expected to exceed supply, the local router tries to find alternative paths to detour
- In the meantime traffic in excess (if any) is cached locally



Three-phase operation

Cache & Detour phase

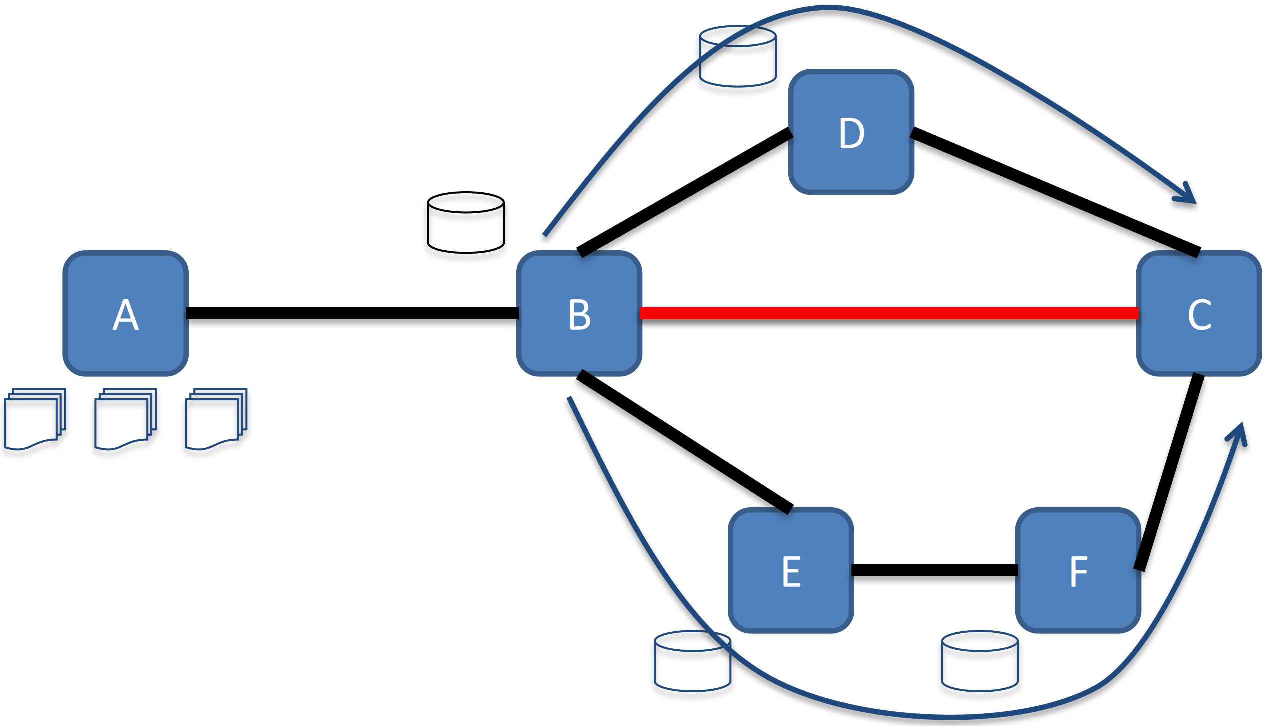
- Every router monitors rate of incoming *Requests*
- When demand is expected to exceed supply, the local router tries to find alternative paths to detour
- In the meantime traffic in excess (if any) is cached locally



Three-phase operation

Cache & Detour phase

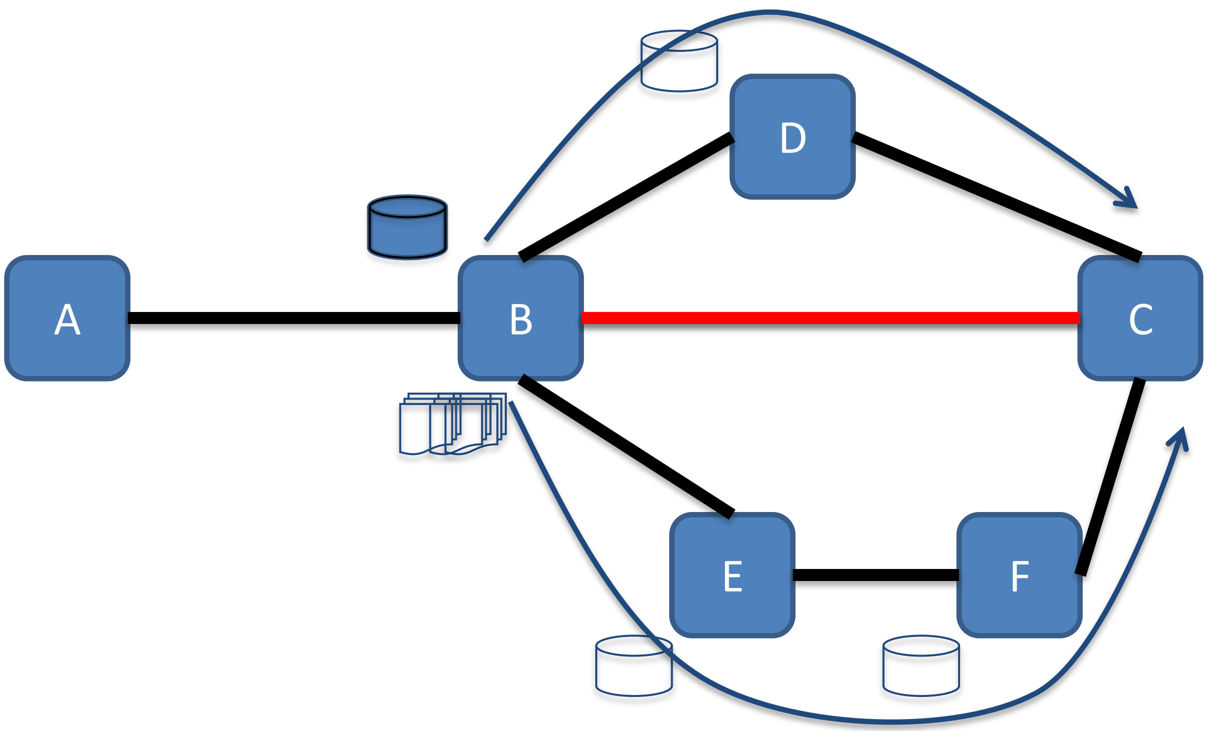
- Every router monitors rate of incoming *Requests*
- When demand is expected to exceed supply, the local router tries to find alternative paths to detour
- In the meantime traffic in excess (if any) is cached locally



Three-phase operation

Cache & Detour phase

- Every router monitors rate of incoming *Requests*
- When demand is expected to exceed supply, the local router tries to find alternative paths to detour
- In the meantime traffic in excess (if any) is cached locally

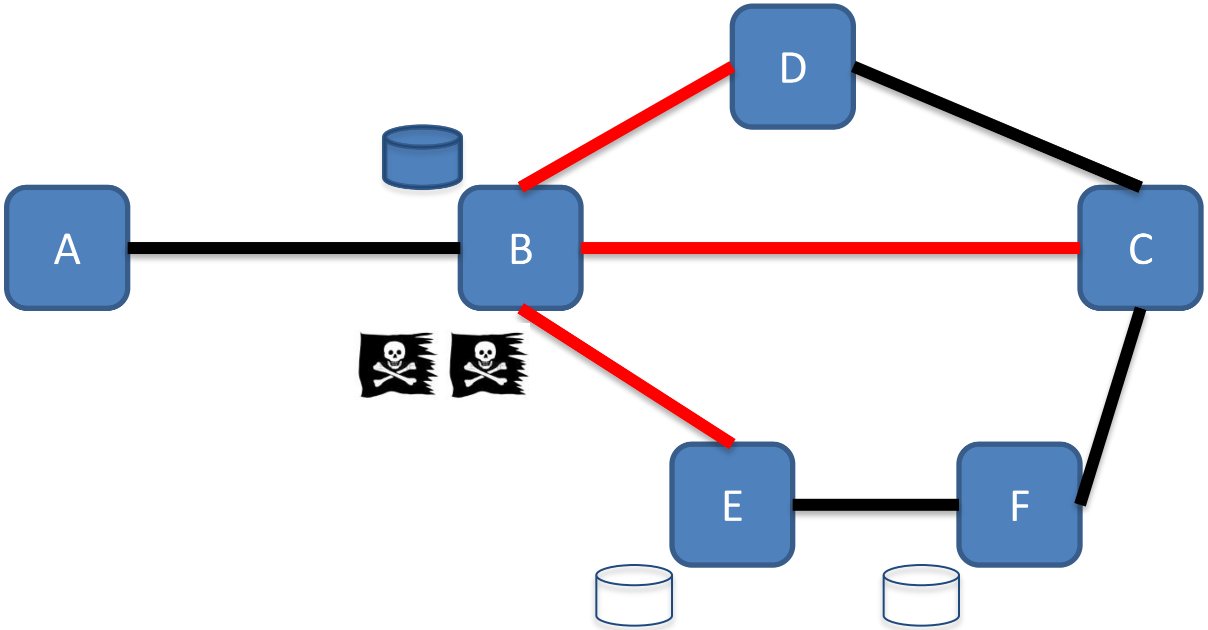


Three-phase operation

Backpressure phase – Closed-loop system

If alternative paths do not exist or are equally congested:

- Pace requests
- Send notification upstream to slow down and enter closed-loop transmission

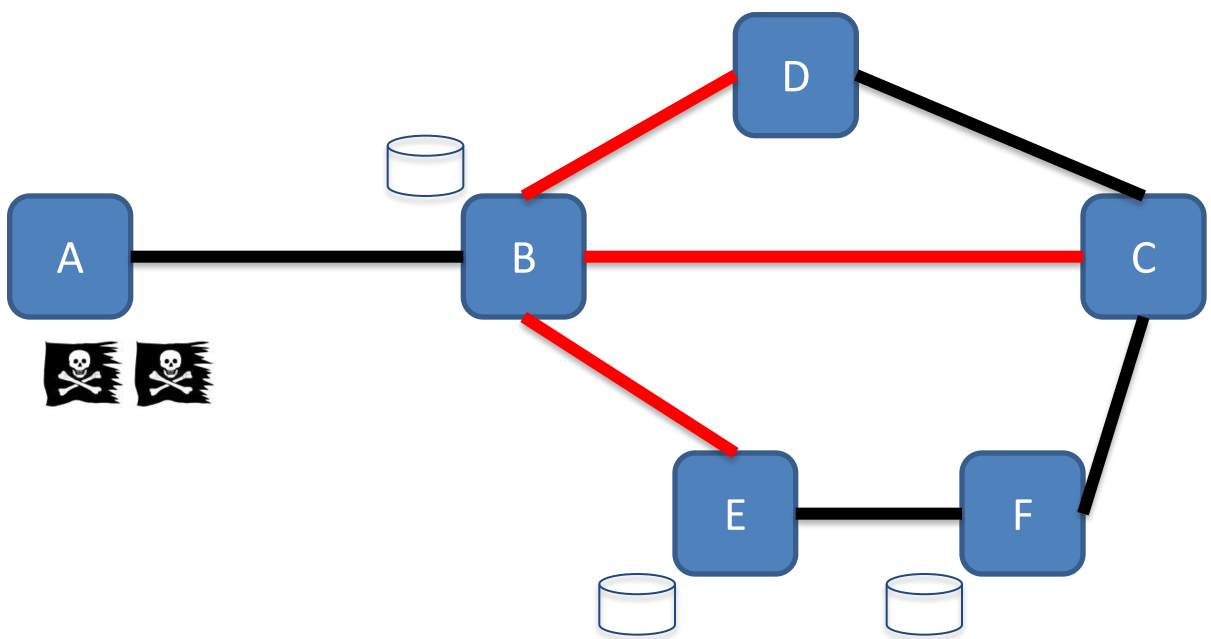


Three-phase operation

Backpressure phase – Closed-loop system

If alternative paths do not exist or are equally congested:

- Pace requests
- Send notification upstream to slow down and enter closed-loop transmission



Feasibility

Feasibility

- Recent work [CuckooSwitch, CoNEXT'13], [MICA, NSDI'14], [Caesar, ANCS'14] shows the feasibility of per-packet read/write operations in DRAM at several 10s Gbps on off-the-shelf hardware.

Feasibility

- Recent work [CuckooSwitch, CoNEXT'13], [MICA, NSDI'14], [Caesar, ANCS'14] shows the feasibility of per-packet read/write operations in DRAM at several 10s Gbps on off-the-shelf hardware.
- DRAM cost is steadily decreasing. We'll soon have TBs of DRAM available on commodity servers.

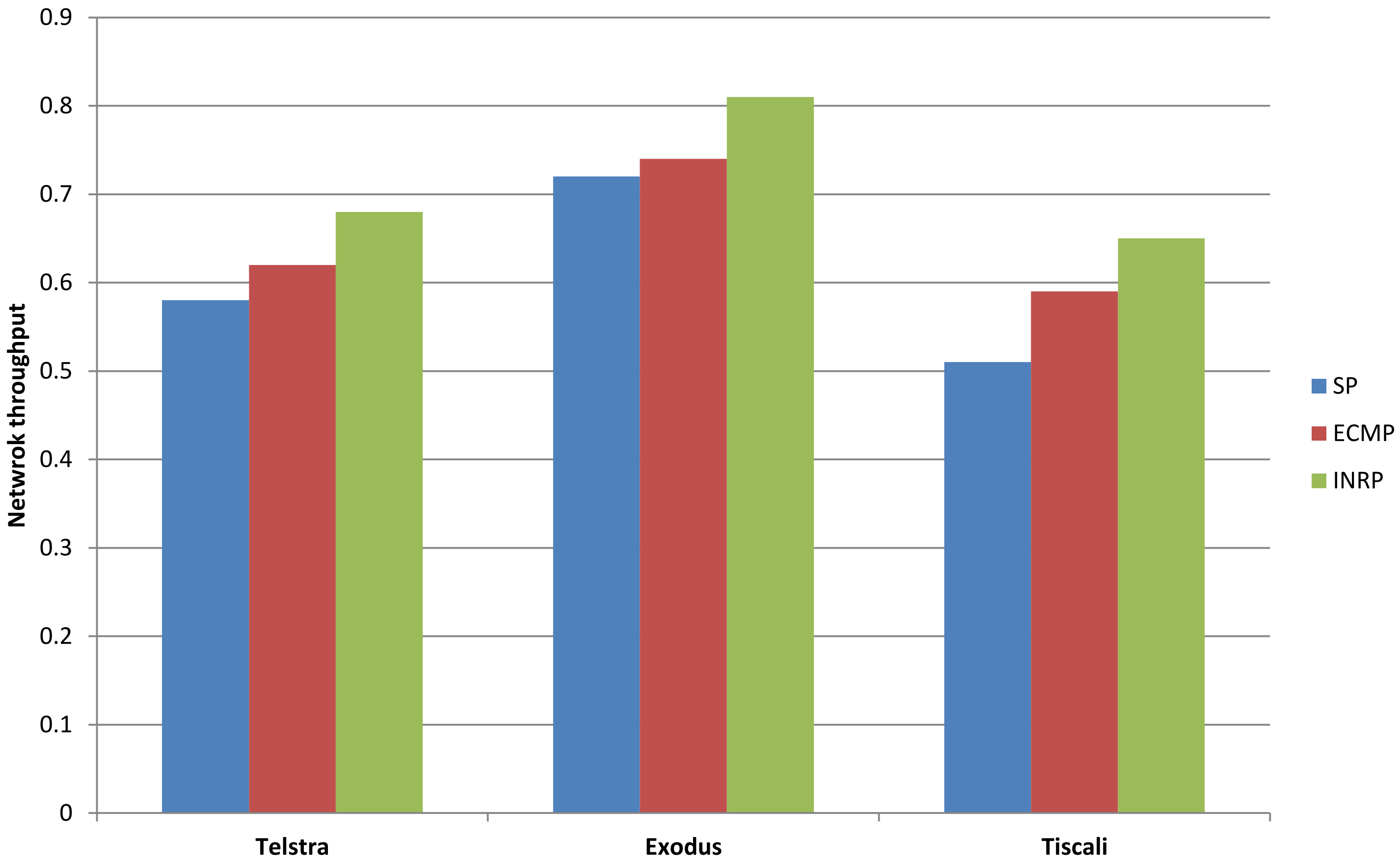
Feasibility

- Recent work [CuckooSwitch, CoNEXT'13], [MICA, NSDI'14], [Caesar, ANCS'14] shows the feasibility of per-packet read/write operations in DRAM at several 10s Gbps on off-the-shelf hardware.
- DRAM cost is steadily decreasing. We'll soon have TBs of DRAM available on commodity servers.
- Ongoing work suggests also Flash-based packet caches could be a viable solution

Availability of detour paths

ISP	1 hop	2 hops	3+ hops	N/A
Exodus (US)	49.77%	35.48%	6.68%	8.06%
VSNL (IN)	25.00%	33.33%	0.00%	41.67%
Level 3	92.22%	6.55%	0.68%	0.55%
Sprint (US)	56.66%	37.08%	1.81%	4.45%
AT&T (US)	34.84%	61.69%	0.72%	2.74%
EBONE (EU)	50.66%	36.22%	6.30%	6.82%
Telstra (AUS)	70.05%	10.42%	1.06%	18.47%
Tiscali (EU)	24.50%	39.85%	10.15%	25.50%
Verio (US)	71.50%	17.09%	1.74%	9.68%
Average	52.80%	30.86%	3.24%	13.10%

Some (very initial) results



Summary and open issues

- Information-Centric Networking:
  - Lots of attention lately
  - Requires investment and effort
  - Worth doing, but need to get the full set of advantages
- There is an opportunity to deal with congestion control **at the network layer**

Summary and open issues

- Information-Centric Networking:
  - Lots of attention lately
  - Requires investment and effort
  - Worth doing, but need to get the full set of advantages
- There is an opportunity to deal with congestion control **at the network layer**
- Open Issues:
  - How do you know detour paths are not congested
  - How will this co-exist with traditional TCP flows?
  - Out of order delivery
  - Flows swapping between original and detour paths